



Stored Procedures Part - II

- Jayendra Khatod

Objectives

- **Describe methods of passing parameters to a procedures**
- **DEFAULT Option for parameters**
- **Invoking Procedure from Anonymous Block**
- **Invoking Procedure from another Procedure**
- **Handled and Unhandled Exceptions**
- **Removing Procedures**

Methods for Passing Parameters

Positional: List actual parameters in the same order as formal parameters.

Named: List actual parameters in arbitrary order by associating each with its corresponding formal parameter.

Combination: List some of the actual parameters as positional and some as named.

DEFAULT Option for Parameters

```
CREATE OR REPLACE PROCEDURE add_dept  
  (p_name IN dept.dname%TYPE  
    DEFAULT 'unknown',  
  p_loc IN dept.loc%TYPE DEFAULT 1700)  
IS  
BEGIN  
  INSERT INTO dept(deptno,dname, loc)  
  VALUES (dept_seq.NEXTVAL, p_name,  
    p_loc);  
END add_dept;  
/
```

Examples of Passing Parameters

BEGIN

add_dept;

add_dept ('TRAINING', 2500);

**add_dept (p_loc => 2400, p_name
=>'EDUCATION');**

add_dept (p_loc => 1200) ;

END;

SELECT deptno, dname, loc

FROM dept;

Declaring Subprograms

```
CREATE OR REPLACE PROCEDURE leave_emp2  
(p_id IN emp.empno%TYPE)  
IS  
    PROCEDURE log_exec  
    IS  
    BEGIN  
        INSERT INTO log_table (user_id, log_date)  
        VALUES (USER, SYSDATE);  
    END log_exec;  
BEGIN  
    DELETE FROM emp  
    WHERE empno = p_id;  
    log_exec;  
END leave_emp2;
```

Invoking a Procedure from an Anonymous Block

DECLARE

v_id NUMBER := 163;

BEGIN

raise_salary(v_id); --invoke procedure

COMMIT;

...

END;

Invoking a Procedure from Another Procedure

```
CREATE OR REPLACE PROCEDURE  
process_emps  
IS  
CURSOR emp_cursor IS  
SELECT empno  
FROM emp;  
BEGIN  
FOR emp_rec IN emp_cursor  
LOOP  
raise_salary(emp_rec.empno);  
END LOOP;  
COMMIT;  
END process_emps;
```


Handled Exceptions

Calling procedure

```
PROCEDURE  
PROC1 ...  
IS  
...  
BEGIN  
...  
PROC2(arg1);  
...  
EXCEPTION  
...  
END PROC1;
```

Called procedure

```
PROCEDURE  
PROC2 ...  
IS  
...  
BEGIN  
...  
EXCEPTION  
...  
END PROC2;
```

Exception raised

Exception handled

Unhandled Exceptions

Calling procedure

```
PROCEDURE  
PROC1 ...  
  IS  
  ...  
  BEGIN  
  ...  
  PROC2(arg1);  
  ...  
  EXCEPTION  
  ...  
END PROC1;
```

Called procedure

```
PROCEDURE  
PROC2 ...  
  IS  
  ...  
  BEGIN  
  ...  
  EXCEPTION  
  ...  
END PROC2;
```

Exception raised

Exception
unhandled

Removing Procedures

Drop a procedure stored in the database.

Syntax:

```
DROP PROCEDURE procedure_name
```

Example

```
DROP PROCEDURE raise_salary;
```

Summary

- **Parameters can be passed by three methods as Named, Positional and a combination of both.**
- **Procedure can contain another procedure as a local sub program.**
- **You should be aware of the effect of handled and unhandled exceptions on transactions and calling procedures.**
- **You can remove procedures from the database by using the DROP PROCEDURE command.**



Thank You !